

Date⇒ 24-02-2021

Module⇒ Backend

Lecture By⇒ Akash Handa

Subject ⇒Crud using Postgres

IN PREVIOUS LECTURE (QUICK RECAP) Date-23/02/2021	In Today's Lecture (Overview)
Values In Postgres Delete In post Group By	Crud Using Postgres App testing using Chai

Crud Using Postgres

```
const express = require('express');
const bodyParser = require('body-parser');
const app = express()
const port = 3000;

app.use(bodyParser.json());
app.use(bodyParser.urlencoded({extended:true}));

const Pool = require('pg').Pool;
const pool = new Pool({
  host:'localhost',
  user:'postgres',
  database:'postgres',
  port:5432
})

app.get('/users', (req, res) => {
  pool.query('SELECT * FROM customers order by id DESC', (err, result) =>{
    if(err) throw err;
    res.send(result.rows)
  })
})
```

```

    })
  })

app.post('/addUser', (req, res) => {
  const {first_name, last_name, gender, phone, email} = req.body
  pool.query('Insert into customers (first_name, last_name, gender,
phone, email) values
($1,$2,$3,$4,$5)', [first_name, last_name, gender, phone, email], (err, result)
=> {
    if(err) throw err;
    res.send("Data Added")
  })
})

app.put('/updateUser', (req, res) => {
  const {first_name, last_name, gender, phone, email, id} = req.body
  pool.query('UPDATE customers SET first_name=
$1, last_name=$2, gender=$3, phone=$4, email=$5 where
id=$6', [first_name, last_name, gender, phone, email, id], (err, result) => {
    if(err) throw err;
    res.send("Data updated")
  })
})

app.delete('/deleteUser', (req, res) => {
  const {id} = req.body
  pool.query('DELETE FROM customers where id=$1', [id], (err, result) => {
    if(err) throw err;
    res.send("Data Deleted")
  })
})

app.listen(port)

```

App testing using Chai

```

let chai = require('chai');
let chaiHttp = require('chai-http');

```

```
let expect = chai.expect
chai.use(chaiHttp)

describe('Testing Rest Api', () => {
  it('should return 200 for health', (done) => {
    chai.request(`http://localhost:9700`)
      .get('/')
      .then((res) => {
        expect(res).to.have.status(200);
        done();
      })
      .catch((err) => {
        throw err
      })
  })
})
})
```

